Ranking Websites, a Probabilistic View

Ying Bao, Gang Feng, Tie-Yan Liu, Zhi-Ming Ma, and Ying Wang

Abstract

In this paper we suggest evaluating the importance of a website with the mean frequency of visiting the website for the Markov chain on the Internet Graph describing random surfing. We show that this mean frequency is equal to the sum of the PageRanks of all the webpages in that website (hence is referred to as PageRankSum), and propose a novel algorithm 'AggregateRank' based on the theory of stochastic complement to calculate the rank of a website. The AggregateRank Algorithm gives a good approximation of the PageRankSum accurately, while the corresponding computational complexity is much lower than PageRankSum. By constructing return-time Markov chains restricted to each website, we describe also the probabilistic relation between PageRank and AggregateRank. The complexity of the AggregateRank Algorithm, the error bound of the estimation, and experiments are discussed at the end of the paper.

Keywords: Website, Web search and mining, PageRank, AggregateRank, Markov chain.

1 Introduction

The design of Web search engine has been becoming a focus of the research on the Web search and mining. One popular aspect is to calculate Static Rank by exploiting the hyperlink structure of the Web. Researchers made great progresses on link analysis models and algorithms since 1998, such as HITS and PageRank [Kleinberg 99, Brin et al. 99]. Nowadays, PageRank has emerged as a popular link analysis model, mostly due to its query-independence, using only the web graph structure, and Google's huge business success.

While a webpage, an essential element of the Web, is still a focus of the research on the Web search and mining, in recent years many researchers have realized that the website, another element of the Web, has played a more and more important role in the Web search and mining

applications[Despeyroux 04, Bavulcu et al. 04, Lerman et al. 04, Lei et al. 04, Qin et al. 05]. As compared to an individual webpage, a website can sometimes provide plenty of semantic information in the following aspects. First, webpages in the same website may be authored by the same person or organization. Second, there might be high correlations between webpages in the same website, in terms of content, page layout and hyperlinks. Third, from the topological point of view, websites contain higher density of hyperlinks inside them (about 75% according to [Henzinger et al. 03]) and lower density of edges in between[Girvan and Newman 02]. These properties make websites semantically important to understand the whole picture of the Web. Actually, the ranking of websites has been a key technical component in many commercial search engines. On the one hand, in the service part, it can help define a more reasonable rank for webpages because those pages from important websites tend to be important as well. On the other hand, in the crawler part, it can help crawl webpages from those important websites first, or help determine a quota (number of webpages) for each website in the index according to its rank. In this way, with the same size of index in total, search engines can achieve the best tradeoff between index coverage and index quality.

In the literature of website ranking, researchers used to describe the inter-connectivity among websites with a so-called HostGraph in which the nodes denote websites and the edges denote linkages between websites (there will be an edge between two websites if and only if there are hyperlinks from the webpages in one website to the webpages in the other), and then adopted the random walk model in the HostGraph to calculate the website ranking[Bharat et al. 01, Dill et al. 01]. However, we want to point out that the random walk over such a HostGraph does not reasonably reflect the browsing behavior of web surfers. In this paper we shall propose a reasonable evaluation for ranking the Websites. Namely, we suggest evaluating the importance of a website with the mean frequency of visiting the website for the Markov chain on the Internet Graph describing random surfing. We shall prove (see Theorem 3) that this mean frequency is equal to the sum of the PageRanks of all the webpages in that website (denoted by PageRankSum for ease of reference).

However, it is clear that using PageRankSum to calculate the ranks of websites is not yet a feasible solution, especially for those applications that only care about the websites. The reason is that the number of webpages is much larger than the number of websites. Therefore, it is much more complex to rank web pages than to rank websites, and it is almost impossible for small research groups or companies to afford such kind of expensive computations. To tackle these aforementioned problems, we propose a novel algorithm based on the theory of stochastic complement [Meyer 89] to calculate the rank of a website that can approximate the PageRankSum accurately, while the corresponding computational complexity is much lower than PageRankSum and only a little higher than those previous HostRank algorithms [Bharat

et al. 01, Dill et al. 01]. We name this algorithm AggregateRank. Experiments demonstrated the effectiveness and efficiency of this algorithm.

Since PageRank reflects the mean frequency of visiting webpages (cf. Section 2 below) and AggregateRank reflects the mean frequency of visiting the websites, both algorithms are tightly related. By constructing return-time Markov chains restricted to each website, we may formulate the relation between PageRank and AggregateRank as follows. Suppose that AggregateRank $\xi = (\xi_1, \xi_2, \cdots, \xi_N)$, $P_{S_i}(\alpha)$ denotes the transition matrix of the return-time Markov chain for website S_i (for $i = 1, 2, \cdots, N$) and the stationary distribution of $P_{S_i}(\alpha)$ is $\pi_{S_i}(\alpha)$, $i = 1, 2, \cdots, N$. Then

$$PageRank = (\xi_1 \pi_{S_1}(\alpha), \xi_2 \pi_{S_2}(\alpha), \cdots, \xi_N \pi_{S_N}(\alpha)). \tag{1}$$

The rest of this paper is organized as follows. In Section 2 we briefly review the probabilistic meaning of PageRank and explain that PageRank reflects mean frequency of visiting webpages. In Section 3 we explore how to reasonably rank websites with the mean frequency of visiting it as well, and describe our AggregateRank algorithm. In Section 4 we describe the probabilistic relation between PageRank and AggregateRank. In Section 5 we discuss the complexity and the error bound of AggregateRank Algorithm, and report experiments with some real web graphs data.

2 Explaining PageRank with Markov Chain

How to rank web pages has been investigated widely and one of the most famous algorithms is called PageRank [Brin et al. 99, Langville and Meyer 04], which is proposed by Brin and Page in 1998 and used by Google search engine. The probabilistic meaning of PageRank has been explained in the literature (see e.g. [Langville and Meyer 04]). For the purpose of our further discussion, we briefly review the probabilistic meaning of PageRank and provide a more explicit explanation via ergodic theorem of Markov chains.

Consider the Hyperlink structure of webpages on a network as a directed graph $G = (V(G), E(G))$ [Bao and Liu 06]. A vertex $i \in V(G)$ of the graph represents a webpage and a directed edge $\vec{ij} \in E(G)$ represents a hyperlink from the webpage i to j. Let B be the adjacent matrix of G and b_i be the sum of the i^{th} row of B. Let D be the diagonal matrix with diagonal entry b_i (if $b_i = 0$, then we normalize $b_i = n$, the cardinal number of $V(G)$, and change all entries of the i^{th} row of B to 1). Now, we construct a stochastic matrix $P = D^{-1} \cdot B$.

When a surfer browses on the Internet, he may choose the next page by randomly clicking one of the links in the current page with a large probability α, which means he randomly walks on G with transition probability P. Sometimes, he may open a new page randomly not along

the hyperlinks with a small probability $(1-\alpha)$, which means he randomly walks on G with transition probability $\frac{1}{n} \cdot ee^T$, where e is a column vector of all ones. So, the transition matrix which describes the random surfer behavior is formulated as

$$P(\alpha) = \alpha P + (1-\alpha) \cdot \frac{1}{n} \cdot ee^T. \tag{2}$$

The random surfer model can be formally described by a Markov chain $\{X_k\}_{k\geq 0}$. The evolution of the Markov chain represents the surfing behavior of a random surfer from one webpage to another. So, the transition matrix of $\{X_k\}_{k\geq 0}$ is $P(\alpha)$, which is an irreducible stochastic matrix on a finite state space and has unique stationary distribution.

PageRank algorithm use the stationary distribution of $P(\alpha)$ (denoted by $\pi(\alpha)$, which satisfies $\pi(\alpha)P(\alpha) = \pi(\alpha)$ with $\pi(\alpha)e = 1.$) to evaluate the importance of webpages. That is to say, webpages are ranked according to their value in $\pi(\alpha)$.

Now we will explain the probabilistic meaning for PageRank more explicitly. We learn from the ergodic theorem on Markov chains (cf e.g. [Qian and Gong 97]) that

$$\pi^i(\alpha) = \lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} p_{ii}^{(k)}(\alpha) \tag{3}$$

$$= \big(\sum_{n=1}^{\infty} n f_{ii}(n)\big)^{-1}, \tag{4}$$

where $\pi^i(\alpha)$ is the i^{th} entry of $\pi(\alpha)$, $p_{ii}^{(k)}(\alpha)$ is the ii^{th} entry of the k-step transition matrix $P^k(\alpha)$ and $f_{ii}(n)$ is the probability of first returning to the page i with n-step after starting from the page i. $\big(\sum_{n=1}^{\infty} n f_{ii}(n)\big)^{-1}$ is equal to the mean frequency of visiting webpage i. The more important a webpage is, the higher frequency it will be visited. So, the ergodic theorem on Markov chains shows that the stationary distribution $\pi(\alpha)$ of $P(\alpha)$ is a very suitable candidate for ranking the webpages.

3 Rank for WebSites in Probabilistic View

PageRank has been proved to be successful in Web search. Actually, ranking is not only important for webpages, but also important for websites in many applications. There were two approaches in the literature of website ranking. However, we will show that the traditional approaches on calculating website ranks are not reasonable because they lose some transition information of the random surfer (see Subsection 3.1). To tackle this problem, we will investigate the real transition probability between websites in Subsection 3.2, and then based on the investigation propose a novel algorithm for website ranking in Subsection 3.3.

3.1 Traditional Approaches to Calculating Website Ranks

In the literature of website ranking, people used to apply those technologies proposed for ranking web pages to the ranking of websites. For example, the famous PageRank algorithm was used to rank websites in [Eiron et al. 04] and [Wu and Aberer 04]. In order to apply PageRank to the ranking of websites, a HostGraph was constructed in these works. In the HostGraph, the nodes denote websites and there is an edge between two nodes if there are hyperlinks from the webpages in one website to the webpages in the other. According to different definitions of the edge weights, two categories of HostGraphs were used in the literature. In the first category, the weight of an edge between two websites was defined by the number of hyperlinks between the two sets of web pages in these sites [Bharat et al. 01]. In the second category, the weight of any edge was simply set to 1 [Dill et al. 01]. For the sake of clarity, we refer to the two categories as weighted HostGraph and *naïve* HostGraph respectively. Figure 1 and 2 show how these two categories of HostGraphs can be constructed.

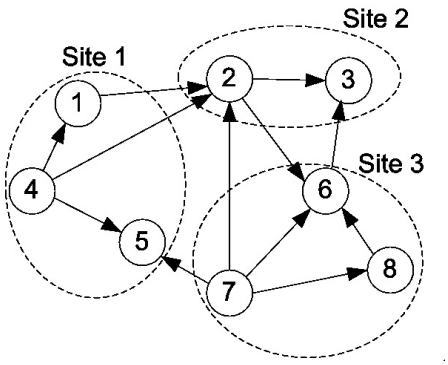

Figure 1: Illustration for web graph and website.

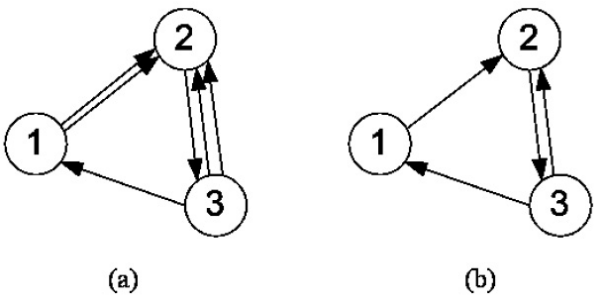

Figure 2: Illustration for (a) weighted HostGraph and (b) *naïve* HostGraph

After constructing the HostGraph, the similar random walk was conducted. That is, a random surfer was supposed to jump between websites following the edges with a probability of α, or jump to a random website with a probability of $1 - \alpha$. In such a way, one can obtain the HostRank, which is used to describe the importance of websites.

At the first glance, the above random walk model over the HostGraph seems to be a natural extension of the PageRank algorithm. However, we want to point out that it is actually not as reasonable as PageRank because it is not in accordance with the browsing behavior of the Web surfers. As we know, real-world web surfers usually have two basic ways to access the web. One is to type URL in the address edit of the web browser. And the other is to click any hyperlink in the current loaded page. These two manners can be well described by the parameter α used in PageRank. That is, with a probability of $1 - \alpha$, the web users visit a random web page by inputting its URL (using favorite folder can also be considered as a shortcut of typing URL), and with a probability of α, they visit a web page by clicking a hyperlink.

Nevertheless, as for the random walk in the HostGraph, we can hardly find the same evident correlation between the random walk model and real-world user behaviors. For example, even if there is a edge between two websites A and B in the HostGraph, when a web surfer visits a page in website A, he may not be able to jump to website B because the hyperlink to website B may exist in another page in website A which is even unreachable from the page that he is currently visiting. In other words, the HostGraph is only a kind of approximation to the web graph: it loses much transition information, especially as for the *naïve* HostGraph. As a result, we argue that the rank values derived from the aforementioned HostGraph are not convincing enough.

3.2 Transition Probability between Websites

Motivated by the probabilistic explanation of PageRank discussed in section 2, we propose that a reasonable way to describe the importance of a website should be the mean frequency that the users visit it.

Actually, the Markov chain $\{X_k\}_{k\geq 0}$ also implies the transition between websites. We should do a little deduction to expose it.

Suppose there are totally N websites in the Web. As each webpage belongs to some determinate website, we rearrange the transition matrix $P(\alpha)$ and partition it into $N \times N$ blocks

according to the N websites. Then it has the following form

$$P(\alpha) = \begin{pmatrix} P_{11}(\alpha) & P_{12}(\alpha) & \cdots & P_{1N}(\alpha) \\ P_{21}(\alpha) & P_{22}(\alpha) & \cdots & P_{2N}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1}(\alpha) & P_{N2}(\alpha) & \cdots & P_{NN}(\alpha) \end{pmatrix}, \tag{5}$$

where the elements in each diagonal block denote the transition probabilities between webpages in the same website, and the elements of each off-diagonal block denote the transition probabilities between webpages in different websites. The diagonal blocks $P_{ii}(\alpha)$ are square and of order n_i, for $i = 1, 2, \cdots, N$, and $n = \sum_{i=1}^{N} n_i$. The stationary distribution $\pi(\alpha)$, known as the PageRank vector, is given by

$$\pi(\alpha)P(\alpha) = \pi(\alpha) \text{ with } \pi(\alpha)e = 1. \tag{6}$$

Let $\pi(\alpha)$ be partitioned conformally with $P(\alpha)$, i.e.,

$$\pi(\alpha) = (\pi_1(\alpha), \pi_2(\alpha), \cdots, \pi_N(\alpha)), \tag{7}$$

and $\pi_i(\alpha)$ is a row vector of length n_i.

Till now, we just get a rearranged PageRank. However, this process is necessary to describe the next part plainly.

We now turn our attention to the mean frequency that a random surfer visits the website S_j (for any fixed $j = 1, \cdots, N$). Since we are interested in the situation that the surfing Markov chain $\{X_k\}_{k\geq 0}$ has run a long time, therefore we may assume that $\{X_k\}_{k\geq 0}$ starts from the stationary probability $\pi(\alpha)$. Then, the one-step transition probability from the website S_i to the website S_j is defined by

$$c_{ij}(\alpha) = Pr_{\pi(\alpha)}\{X_{m+1} \in S_j \mid X_m \in S_i\}. \tag{8}$$

The k-step transition probability from the website S_i to the website S_j is defined by

$$c_{ij}^{(k)}(\alpha) = Pr_{\pi(\alpha)}\{X_{m+k} \in S_j \mid X_m \in S_i\}. \tag{9}$$

Recall that $\|\cdot\|_1$ is the 1-norm of a vector, i.e., the sum of all entries of a vector.

Theorem 3.1. $c_{ij}(\alpha) = \frac{\pi_i(\alpha)}{\|\pi_i(\alpha)\|_1} P_{ij}(\alpha)e$.

Proof. By the properties of conditional probability, we have

$$
\begin{aligned}
c_{ij}(\alpha) &= Pr_{\pi(\alpha)}\{X_{m+1} \in S_j \mid X_m \in S_i\} \\
&= \frac{Pr_{\pi(\alpha)}\{X_{m+1} \in S_j, X_m \in S_i\}}{Pr_{\pi(\alpha)}\{X_m \in S_i\}} \\
&= \frac{\sum_{t \in S_j} \sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_{m+1} = t, X_m = l\}}{\sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_m = l\}} \\
&= \frac{\sum_{t \in S_j} \sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_m = l\} Pr_{\pi(\alpha)}\{X_{m+1} = t \mid X_m = l\}}{\sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_m = l\}} \\
&= \frac{\sum_{t \in S_j} \sum_{l \in S_i} \pi^l(\alpha) p_{lt}(\alpha)}{\sum_{l \in S_i} \pi^l(\alpha)} \\
&= \frac{\pi_i(\alpha) P_{ij}(\alpha) e}{\parallel \pi_i(\alpha) \parallel_1} \\
&= \frac{\pi_i(\alpha)}{\parallel \pi_i(\alpha) \parallel_1} P_{ij}(\alpha) e,
\end{aligned}
$$

where $\pi^l(\alpha)$ is the l^{th} entry of $\pi(\alpha)$, $p_{lt}(\alpha)$ is the lt^{th} entry of $P(\alpha)$, and e is a column vector of all ones of which the dimension depends on the corresponding context. □

Theorem 3.2. $c_{ij}^{(k)}(\alpha) = \frac{\pi_i(\alpha)}{\|\pi_i(\alpha)\|_1} P_{ij}^{(k)}(\alpha) e$, *where* $P_{ij}^{(k)}(\alpha)$ *is the* ij^{th} *block of the* k*-step transition matrix* $P^k(\alpha)$.

Proof. By the properties of conditional probability, we have

$$
\begin{aligned}
c_{ij}^{(k)}(\alpha) &= Pr_{\pi(\alpha)}\{X_{m+k} \in S_j \mid X_m \in S_i\} \\
&= \frac{Pr_{\pi(\alpha)}\{X_{m+k} \in S_j, X_m \in S_i\}}{Pr_{\pi(\alpha)}\{X_m \in S_i\}} \\
&= \frac{\sum_{t \in S_j} \sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_{m+k} = t, X_m = l\}}{\sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_m = l\}} \\
&= \frac{\sum_{t \in S_j} \sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_m = l\} Pr_{\pi(\alpha)}\{X_{m+k} = t \mid X_m = l\}}{\sum_{l \in S_i} Pr_{\pi(\alpha)}\{X_m = l\}} \\
&= \frac{\sum_{t \in S_j} \sum_{l \in S_i} \pi^l(\alpha) p_{lt}^{(k)}(\alpha)}{\sum_{l \in S_i} \pi^l(\alpha)} \\
&= \frac{\pi_i(\alpha) P_{ij}^{(k)}(\alpha) e}{\parallel \pi_i(\alpha) \parallel_1} \\
&= \frac{\pi_i(\alpha)}{\parallel \pi_i(\alpha) \parallel_1} P_{ij}^{(k)}(\alpha) e,
\end{aligned}
$$

where $p_{lt}^{(k)}(\alpha)$ is the lt^{th} entry of $P^k(\alpha)$. □

We have assumed that $\pi(\alpha)$ is the initial distribution of the webpage surfing Markov chain $\{X_k\}_{k\geq 0}$. We assume that a surfer is browsing on some website S_i at time m, and we will calculate the number of visiting the website S_j from now on. Let $N_j(n)$ denote the number of $\{X_k\}_{k\geq 0}$ visiting the website S_j during the n times $\{m+1, m+2, \cdots, m+n\}$. Then we can get the following conclusion.

Theorem 3.3. $\| \pi_j(\alpha) \|_1 = E\big(\lim_{n\to\infty} \frac{N_j(n)}{n}\big)$.

Proof. By definition, we know that $E\big(\lim_{n\to\infty} \frac{N_j(n)}{n}\big)$ is the mean frequency of visiting the website S_j. Hence, by dominated convergence theorem (e.g. [Kallenberg 97]) and the ergodic theorem on Markov chains, we get

$$\begin{aligned}
& E\big(\lim_{n\to\infty} \frac{N_j(n)}{n}\big) \\
= \; & E\big(\lim_{n\to\infty} \frac{\sum_{k=1}^{n} \mathbf{1}_{\{\mathbf{X_{m+k}}\in \mathbf{S_j}\}}}{n}\big) \\
= \; & \lim_{n\to\infty} E\big(\frac{\sum_{k=1}^{n} \mathbf{1}_{\{\mathbf{X_{m+k}}\in \mathbf{S_j}\}}}{n}\big) \\
= \; & \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} c_{ij}^{(k)}(\alpha) \\
= \; & \lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} \frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} P_{ij}^{(k)}(\alpha) e \\
= \; & \frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} (\lim_{n\to\infty} \frac{1}{n}\sum_{k=1}^{n} P_{ij}^{(k)}(\alpha)) e \\
= \; & \frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} (e\pi_j(\alpha)) e \\
= \; & \| \pi_j(\alpha) \|_1,
\end{aligned}$$

where

$$\mathbf{1}_{\{\mathbf{X_{m+k}}\in \mathbf{S_j}\}} = \begin{cases} 1, & \text{when } X_{m+k} \in S_j; \\ 0, & \text{otherwise.} \end{cases}$$

□

From the deduction above, we know that $\| \pi_j(\alpha) \|_1$ is the mean frequency of visiting the website S_j. Hence the probability vector $(\| \pi_1(\alpha) \|_1, \| \pi_2(\alpha) \|_1, \cdots, \| \pi_N(\alpha) \|_1)$ is a suitable candidate for ranking the importance of websites.

As aforementioned, $c_{ij}(\alpha)$ represents the transition probability between websites. By virtue of Theorem 3.1 we see that the $N \times N$ matrix $C(\alpha) = (c_{ij}(\alpha))$ is equal to the coupling matrix

specified in Theorem 4.1 of [Meyer 89]. It follows that $C(\alpha)$ is an irreducible stochastic matrix, so that it possesses a unique stationary probability vector, denoted by $\xi(\alpha)$, i.e.,

$$\xi(\alpha)C(\alpha) = \xi(\alpha) \text{ with } \xi(\alpha)e = 1. \tag{10}$$

One can easily verify that if we define

$$\xi(\alpha) = (\parallel \pi_1(\alpha) \parallel_1, \parallel \pi_2(\alpha) \parallel_1, \cdots, \parallel \pi_N(\alpha) \parallel_1), \tag{11}$$

then $\xi(\alpha)$ is a solution of (10).

One may have realized that the above computation can also be regarded as being carried out with a certain HostGraph. However, the edge weight of this new HostGraph is not decided heuristically as in previous works [Bharat et al. 01, Dill et al. 01], but determined by a sophisticated formulation in Theorem 1. Besides, the transition probability from S_i to S_j actually summarizes all the cases that the random surfer jumps from any webpage in S_i to any webpage in S_j within one-step transition. Therefore, the transition in this new HostGraph is in accordance with the real behavior of the Web surfers. In this regard, the so-calculated rank from the coupling matrix $C(\alpha)$ will be more reasonable than those previous works.

Based on the above discussions, the direct approach of computing the AggregateRank $\xi(\alpha)$ is to accumulate PageRank values (denoted by PageRankSum). However, this approach is unfeasible because the computation of PageRank is not a trivial task when the number of web pages is as large as several billions. Therefore, efficient computation becomes a significant problem. In the next subsection, we will propose an approximate algorithm for this purpose, which can be much more efficient than PageRankSum with very little accuracy loss.

3.3 The AggregateRank Algorithm

As aforementioned, the coupling matrix $C(\alpha)$, with $c_{ij}(\alpha)=\frac{\pi_i(\alpha)}{\|\pi_i(\alpha)\|_1}P_{ij}(\alpha)e$, represents the transition probability from one website to the other in terms of random surfer model; and its stationary distribution $\xi(\alpha)$, which is equal to PageRankSum, is regarded as a reasonable rank of websites. It is clear that the construction of the coupling matrix asks for the calculation of PageRank over the whole Web graph. To avoid this time-consuming step, it is necessary to invent a new method to construct the coupling matrix $C(\alpha)$. Fortunately, the theory of stochastic complement [Meyer 89] gives a good solution to form $C(\alpha)$ without PageRank values. Intuitively, the stochastic complement of a diagonal block $P_{ij}(\alpha)$ in (5) represents the transition matrix for the return-time Markov chain (cf. Subsection 4 below) restricted to the webpages of the i-th websites. To illustrate this, we take a simple web graph for example. Suppose the web graph only contains two websites, the transition probability matrix of this

Web graph (rearranged according to website information) can be denoted by

$$P(\alpha) = \begin{pmatrix} P_{11}(\alpha) & P_{12}(\alpha) \\ P_{21}(\alpha) & P_{22}(\alpha) \end{pmatrix}, \tag{12}$$

and its stationary distribution is $\pi(\alpha) = (\pi_1(\alpha), \pi_2(\alpha))$. For each diagonal block in $P(\alpha)$, we can calculate its stochastic complement. For example, the stochastic complement of $P_{11}(\alpha)$ is calculated as follow (see also (36) below),

$$S_{11}(\alpha) = P_{11}(\alpha) + P_{12}(\alpha)(I - P_{22}(\alpha))^{-1}P_{21}(\alpha). \tag{13}$$

The stochastic complement is also a stochastic matrix, each row of which is summed up to 1. It can be proved that $\frac{\pi_1(\alpha)}{\|\pi_1(\alpha)\|_1}$ is the unique stationary probability vector for the stochastic complement $S_{11}(\alpha)$, i.e.

$$\frac{\pi_1(\alpha)}{\| \pi_1(\alpha) \|_1} S_{11}(\alpha) = \frac{\pi_1(\alpha)}{\| \pi_1(\alpha) \|_1} \text{ with } \frac{\pi_1(\alpha)}{\| \pi_1(\alpha) \|_1} e = 1. \tag{14}$$

Generally, $\frac{\pi_i(\alpha)}{\|\pi_i(\alpha)\|_1}$ is the unique stationary distribution for the stochastic complement $S_{ii}(\alpha)$, i.e.

$$\frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} S_{ii}(\alpha) = \frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} \text{ with } \frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} e = 1. \tag{15}$$

Apparently, the computation of the stationary distribution of each $S_{ii}(\alpha)$ will be cheaper than that of PageRank directly because the dimension of each $S_{ii}(\alpha)$ is very small and equal to the page number in this site (we will discuss this in more detail in the next subsection). However, it is time consuming to compute the exact stochastic complement since we should compute a inverse matrix for $I - P_{ii}(\alpha)$. As we know, the computation for inverse matrix is very expensive, which is sometimes even more expensive than PageRank. Thus, we prefer an approximate approach to get the stationary distribution of each stochastic complement instead. According to [Cho and Meyer], we can find an efficient approximate method. It does not use (13) to aggregate the stochastic complement directly. Instead, it only modifies each diagonal block $P_{ii}(\alpha)$ by a little to get a new matrix with the same dimension as $S_{ii}(\alpha)$. The details are given as follows.

For the first step, we modify the original diagonal block $P_{ii}(\alpha)$ to be a transition probability matrix. It is clear that the sum of each row in the original diagonal block $P_{ii}(\alpha)$ is always less than 1. To make it a transition probability matrix, we simply adjust the diagonal elements of $P_{ii}(\alpha)$ (added or subtracted by a small value) to make the sum of each row equal to 1. Letting $P_{ii}^*(\alpha)$ denote the matrix after adjustment, we can calculate its stationary distribution $u_i(\alpha)$ as follows,

$$u_i(\alpha)P_{ii}^*(\alpha) = u_i(\alpha) \text{ with } u_i(\alpha)e = 1. \tag{16}$$

According to [Cho and Meyer], we can prove that (see also (46) below)

$$\frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} \approx u_i(\alpha). \tag{17}$$

From the description above, $P^*_{ii}(\alpha)$ is very easy to get from $P_{ii}(\alpha)$. Moreover, it can even be stored sparsely like original $P(\alpha)$. Thus, formulation (17) means that we can get each $\frac{\pi_1(\alpha)}{\|\pi_1(\alpha)\|_1}$ very efficiently.

Utilizing the result of (17), we can obtain an approximate coupling matrix $C^*(\alpha)$ as below,

$$(C^*(\alpha))_{ij} = u_i(\alpha)P_{ij}(\alpha)e. \tag{18}$$

Consequently, the stationary distribution $\xi^*(\alpha)$ of the approximate coupling matrix can be regarded as a good approximation to $\xi(\alpha)$. We name the aforementioned algorithm by AggregateRank algorithm, whose detailed algorithm flow is shown as follows:

1. Divide the $n \times n$ matrix $P(\alpha)$ into $N \times N$ blocks according to the N sites.
2. Construct the stochastic matrix $P^*_{ii}(\alpha)$ for $P_{ii}(\alpha)$ by changing the diagonal elements of $P_{ii}(\alpha)$ to make each row sum up to 1.
3. Determine $u_i(\alpha)$ from

$$u_i(\alpha)P^*_{ii}(\alpha) = u_i(\alpha) \text{ with } u_i(\alpha)e = 1. \tag{19}$$

4. Form an approximation $C^*(\alpha)$ to the coupling matrix $C(\alpha)$, by evaluating

$$(C^*(\alpha))_{ij} = u_i(\alpha)P_{ij}(\alpha)e. \tag{20}$$

5. Determine the stationary distribution of $C^*(\alpha)$ and denote it $\xi^*(\alpha)$, i.e.,

$$\xi^*(\alpha)C^*(\alpha) = \xi^*(\alpha) \text{ with } \xi^*(\alpha)e = 1. \tag{21}$$

To sum up, the proposed algorithm improves the efficiency in the following ways. First, it uses an easy-to-construct sparse matrix to replace the stochastic complement to approximate $\frac{\pi_i(\alpha)}{\|\pi_i(\alpha)\|_1}$ instead of the whole transition probability matrix. Second, this algorithm is much easier to be implemented in parallel than PageRankSum. The reason is that for PageRank, if we want to implement it in parallel, we must take care of the information exchange between different servers since there are hyperlinks which sources and destinations are not in the same server. While, for our method, we do not need the exchange of information if we put a website at most in one sever since the computations over $P^*_{ii}(\alpha)$ is done for each particular website and is independent of other part of the whole web graph.

4 Probabilistic Relation between PageRank and AggregateRank

As is known, PageRank is carried out with the Webpage Graph where each vertex represents a webpage, while AggregateRank is done with the SiteGraph where each vertex represents a website. If we can discover how the Markov chain $\{X_k\}_{k\geq 0}$ evolves when restricted to webpages of one website, we can reveal the probability relation between PageRank and AggregateRank.

We shall introduce return times to construct a new Markov chain which describes the random surfing behavior on pages of some fixed website A. Assume a starting state in website A, i.e. $X_0 \in A$. The variable

$$\tau_A := \min\{n \geq 1; X_n \in A\} \tag{22}$$

is called the first return time on A. In order to distinguish different return times, we write $\tau_A(k)$ for the random time of the k^{th} visit to A, these are defined inductively by

$$\tau_A(1) := \tau_A, \tag{23}$$

$$\tau_A(k) := \min\{n > \tau_A(k-1); X_n \in A\}. \tag{24}$$

It is clear that the variables $\tau_A(k)$ are stopping times for X.

We construct a new stochastic process $\{\phi_k\}_{k\geq 0}$ as follows:

$$\phi_0 := X_0, \tag{25}$$

$$\phi_k := X_{\tau_A(k)}. \tag{26}$$

Then we can prove that $\{\phi_k\}_{k\geq 0}$ is a time-homogeneous Markov chain, furthermore, an ergodic chain (see[Meyer 89]).

We now turn our attention to the transition probability matrix of $\{\phi_k\}_{k\geq 0}$. Assume that the transition probability matrix $P(\alpha)$ is permuted and repartitioned so that

$$P(\alpha) = \begin{array}{c} \\ A \\ \tilde{A} \end{array} \begin{array}{c} \begin{array}{cc} A & \tilde{A} \end{array} \\ \begin{pmatrix} \tilde{P}_{11}(\alpha) & \tilde{P}_{12}(\alpha) \\ \tilde{P}_{21}(\alpha) & \tilde{P}_{22}(\alpha) \end{pmatrix} \end{array}, \tag{27}$$

where $\tilde{P}_{11}(\alpha)$ denotes the transition probabilities between webpages of website A. Let the stationary distribution $\pi(\alpha)$ be partitioned conformally with $P(\alpha)$, i.e.,

$$\pi(\alpha) = (\pi_1(\alpha), \pi_2(\alpha)). \tag{28}$$

Assume that the website A is composed by webpages $\{a_1, a_2, \cdots, a_m\}$. For the new Markov chain $\{\phi_k\}_{k\geq 0}$, the one-step transition probability of moving from a_k to a_j is the probability

in the original chain $\{X_k\}_{k\geq 0}$ of moving directly from a_k to a_j plus the probability of moving directly from a_k to some state in $\tilde{A}$, and then eventually moving back to A, hitting a_j first upon return [Meyer 89]. The probability of moving directly from a_k to a_j in the original chain $\{X_k\}_{k\geq 0}$ is

$$q_{kj} = [\tilde{P}_{11}(\alpha)]_{kj}, \tag{29}$$

and the probability of moving directly from a_k to $\tilde{a}_h \in \tilde{A}$ is

$$q_{kh} = [\tilde{P}_{12}(\alpha)]_{kh}. \tag{30}$$

The probability of moving from $\tilde{a}_h$ to A such that a_j is the first state entered upon return to A is

$$\begin{aligned} q_{hj} &= \sum_{m\geq 0} [\tilde{P}_{22}(\alpha)^m \tilde{P}_{21}(\alpha)]_{hj} && (31)\\ &= [(I - \tilde{P}_{22}(\alpha))^{-1} \tilde{P}_{21}(\alpha)]_{hj}. && (32) \end{aligned}$$

where $[\tilde{P}_{22}(\alpha)^m \tilde{P}_{21}(\alpha)]_{hj}$ is the probability that the original chain starts from the state $\tilde{a}_h \in \tilde{A}$, runs exactly m times within $\tilde{A}$, and moves from a state in $\tilde{A}$ to the state $a_j \in A$ at the $(m+1)^{th}$ run. q_{hj} can also be obtained by considering the states in A to be absorbing states and applying the theory of absorbing chains, see [Kemeny and Snell 76]. Consequently, the one-step transition probability of moving from a_k to a_j in the new Markov chain $\{\phi_k\}_{k\geq 0}$ is

$$\begin{aligned} & P\{\phi_{n+1} = a_j | \phi_n = a_k\} && (33)\\ = \; & q_{kj} + \sum_{\tilde{A}_h \in \tilde{A}} q_{kh} q_{hj} && (34)\\ = \; & [\tilde{P}_{11}(\alpha) + \tilde{P}_{12}(\alpha)(I - \tilde{P}_{22}(\alpha))^{-1} \tilde{P}_{21}(\alpha)]_{kj}. && (35) \end{aligned}$$

So, the transition matrix of $\{\phi_k\}_{k\geq 0}$, denoted by $P_A(\alpha)$, is formulated as

$$P_A(\alpha) = \tilde{P}_{11}(\alpha) + \tilde{P}_{12}(\alpha)(I - \tilde{P}_{22}(\alpha))^{-1} \tilde{P}_{21}(\alpha). \tag{36}$$

In the theory of Markov chains, $P_A(\alpha)$ is called the stochastic complementation of $\tilde{P}_{11}(\alpha)$ ([Meyer 89, Stewart 94]), it is known that $P_A(\alpha)$ is an irreducible stochastic matrix and has unique stationary distribution. The stationary distribution of $P_A(\alpha)$ is

$$\pi_A(\alpha) = \frac{\pi_1(\alpha)}{\| \pi_1(\alpha) \|_1}. \tag{37}$$

With the above procedure we can construct a return-time Markov chain for each site. We have supposed there are totally N sites and n pages in the Web graph. The transition matrix

$P(\alpha)$ is partitioned into $N \times N$ blocks according to the N sites and has the following form

$$P(\alpha) = \begin{pmatrix} P_{11}(\alpha) & P_{12}(\alpha) & \cdots & P_{1N}(\alpha) \\ P_{21}(\alpha) & P_{22}(\alpha) & \cdots & P_{2N}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1}(\alpha) & P_{N2}(\alpha) & \cdots & P_{NN}(\alpha) \end{pmatrix}, \tag{38}$$

The stationary distribution $\pi(\alpha)$ is partitioned conformally with $P(\alpha)$, i.e.,

$$\pi(\alpha) = (\pi_1(\alpha), \pi_2(\alpha), \cdots, \pi_N(\alpha)), \tag{39}$$

Let $P_{S_i}(\alpha)$ denote the transition matrix of the return-time Markov chain for site S_i (for $i = 1, 2, \cdots, N$). From the deduction above, it is clear that the unique stationary distribution of $P_{S_i}(\alpha)$ is

$$\pi_{S_i}(\alpha) = \frac{\pi_i(\alpha)}{\parallel \pi_i(\alpha) \parallel_1}, \text{ for } i = 1, 2, \cdots, N. \tag{40}$$

We may conclude that the relation between PageRank and AggregateRank can be formulated as follows. Suppose that AggregateRank $\xi = (\xi_1, \xi_2, \cdots, \xi_N)$ and the stationary distribution of $P_{S_i}(\alpha)$ is $\pi_{S_i}(\alpha)$, $i = 1, 2, \cdots, N$. Then

$$PageRank = (\xi_1 \pi_{S_1}(\alpha), \xi_2 \pi_{S_2}(\alpha), \cdots, \xi_N \pi_{S_N}(\alpha)). \tag{41}$$

In other words, the mean frequency of visiting a webpage i which belongs to a website S_j can be decomposed into two factors: one is the mean frequency of visiting the website S_j, the other is the mean frequency of visiting webpage i when restricted to website S_j.

Remark: The above discussions are partly motivated by the stochastic complementation theory and can be regarded as an interesting application of the theory. We refer the reader to [Meyer 89] for a detailed review of stochastic complementation theory.

5 Algorithm Analysis and Experiment

In this section, we will discuss the convergence speed and the error bound of AggregateRank Algorithm. And then, some experiments with the real web graphs are shown. The results of this section has been reported in more detail on the engineering aspects on the 29th Annual International Conference on Research and Development on Information Retrieval, Seattle, 2006 (see[Feng et al. 06]).

5.1 Complexity Analysis

As can be seen in the previous sections, in our proposed AggregateRank algorithm, we divide the Web graph into websites, and conduct power-method iterations within each website. After that, we apply power method once again to the coupling matrix $C^*(\alpha)$. It is easy to understand that, in this way, we can save some memory and the corresponding algorithm is easier to be implemented in parallel. When we deal with the Web graph with billions of pages, this advantage will become very meaningful.

However for the computational complexity, it is not obvious whether the proposed method can be more efficient. The reason is that PageRank has a complexity of $O(r)$ (Suppose there are N sites, n pages, and r hyperlinks in total, $r \approx 10n$). Considering that 75% of the hyperlinks connect pages in the same website [Henzinger et al. 03], dividing the Web graph into websites can only save 25% propagations along hyperlinks and thus the complexity is still around $O(r)$. Furthermore, for the computation in Step 5, it is not obvious whether $C^*(\alpha)$ is also a sparse matrix, thus its computational complexity might be as high as $O(N^2)$ in the worse case. All of this can be a big issue.

In this subsection, we will discuss the aforementioned problems in detail. Specifically, we will prove in Subsection 3.3.2.1 that although the complexity of one iteration of power method applied to $P_{ii}^*(\alpha)$ $(i = 1, \cdots, N)$ is also $O(r)$, its convergence speed can be significantly faster than PageRank over the whole Web graph. And then we will prove in Subsection 3.3.2.2 that $C^*(\alpha)$ is actually also a sparse matrix, and there are only about $O(N)$ non-zero elements in this matrix. Therefore, the computation of calculating stationary distribution for this matrix will also be faster than that for the PageRank matrix.

5.1.1 Convergence Speed Analysis for Power Method Applied to $P_{ii}^*(\alpha)$

In order to understand the convergence speed of the power method applied to $P_{ii}^*(\alpha)$, we need to review the following Lemma at first. As we know, the convergence speed of the power method is determined by the magnitude of the subdominant eigenvalue of the transition probability matrix [Stewart 94]. Lemma 1 just tells us the relationship between matrix $P(\alpha)$ and its eigenvalues [Langville and Meyer 04].

Lemma 1 (**Langville and Meyer**) : *Given the spectrum of the stochastic matrix P as $\{1, \lambda_2, \lambda_3, \cdots, \lambda_n\}$, the spectrum of the primitive stochastic matrix $P(\alpha) = \alpha P + (1-\alpha)ev^T$ is $\{1, \alpha\lambda_2, \alpha\lambda_3, \cdots, \alpha\lambda_n\}$, where v^T is a probability vector.*

In order to use Lemma 1 to analyze the convergence speed of the power method applied to $P_{ii}^*(\alpha)$, we transform $P_{ii}^*(\alpha)$ into the following form

$$P_{ii}^*(\alpha) = \alpha \bar{P}_{ii}^* + (1-\alpha)\frac{1}{n_i} e e^T, \tag{42}$$

where $\bar{P}_{ii}^*$ is a stochastic matrix and e^T/ni is a probability vector.

Given the eigenvalues of $\bar{P}_{ii}^*$ as $\{1, \tilde{\lambda}_2, \tilde{\lambda}_3, \cdots, \tilde{\lambda}_n\}$, by Lemma 1, we can get that the eigenvalues of $P_{ii}^*(\alpha)$ is $\{1, \alpha\tilde{\lambda}_2, \alpha\tilde{\lambda}_3, \cdots, \alpha\tilde{\lambda}_n\}$. Since the convergence speed of the power method is determined by the magnitude of the subdominant eigenvalue of $P_{ii}^*(\alpha)$, we can conclude that the convergence rate of the power method applied to $P_{ii}^*(\alpha)$ is around the rate at which $(\alpha\tilde{\lambda}_2)^k \to 0$.

The convergence speed of PageRank is around the rate at which $\alpha^k \to 0$. So whether we can be more efficient than PageRank is determined by how small $\tilde{\lambda}_2$ could be. According to the following discussions, we know that $\tilde{\lambda}_2 \ll 1$.

As we know, the web link graph has a natural block structure: the majority of hyperlinks are intra-site links [Kamvar et al. 03]. Therefore, the random walk on the web with the transition matrix $P(\alpha)$ can be viewed as a nearly completely decomposable Markov chain. According to [Meyer 89], we know that when the states in a nearly completely decomposable Markov chain are naturally ordered, and when the transition matrix is partitioned into N closely coupled subclasses in the natural way, the underlying transition matrix of the Markov chain has exactly N-1 non-unit eigenvalues clustered near $\lambda = 1$ (There are pathological cases, but they are rare in practical work [Meyer 89]). Thus $P(\alpha)$ has exactly N-1 non-unit eigenvalues clustered near $\lambda = 1$.

Since $P_{ii}^*(\alpha)$ is an irreducible stochastic matrix, the Perron-Frobenius theorem [Gantmacher 59] guarantees that the unit eigenvalue of each $P_{ii}^*(\alpha)$ is simple. Because $P_{ii}^*(\alpha) \approx P_{ii}(\alpha)$, by the continuity of the eigenvalues, the non-unit eigenvalues of $P_{ii}^*(\alpha)$ must be rather far from the unit eigenvalue of $P_{ii}^*(\alpha)$. Otherwise the spectrum of $P_{ii}(\alpha)$ would contain a cluster of at least N non-unit eigenvalues positioned near $\lambda = 1$. As a result, we can come to the conclusion that $\alpha\tilde{\lambda}_2 \ll 1$ for any α close to 1, and then $\tilde{\lambda}_2 \ll 1$. That is, the convergence speed of power method applied to $P_{ii}^*(\alpha)$ is much faster than that of PageRank.

5.1.2 Complexity of Power Method Applied to $C^*(\alpha)$

As mentioned at the beginning of this section, the sparseness of the matrix $C^*(\alpha)$ is a critical factor that influences the computational complexity of our proposed AggregateRank algorithm. To understand this, we conduct the following discussions. First of all, we transform $P(\alpha)$ into

the following form

$$\begin{aligned} P(\alpha) &= \alpha\bar{P} + (1-\alpha)\frac{1}{n}ee^T \\ &= \alpha(P + a\frac{1}{n}e^T) + (1-\alpha)\frac{1}{n}ee^T \\ &= \alpha P + (\alpha a + (1-\alpha)e)\frac{1}{n}e^T, \end{aligned} \tag{43}$$

where P is the transition matrix whose element p_{ij} is the probability of moving from webpage i to webpage j in one step following the hyperlink structure of the Web Graph, a is a vector whose element $a_i = 1$ if row i of P corresponds to a dangling node, and 0, otherwise, α is damping factor, and e is a column vector of all ones. Then we investigate the construction process of $C^*(\alpha)$ as follows,

$$\begin{aligned} C^*(\alpha) &= U(\alpha)P(\alpha)V \\ &= U(\alpha)(\alpha P + (\alpha a + (1-\alpha)e)\frac{1}{n}e^T)V \\ &= \alpha U(\alpha)PV + (\alpha U(\alpha)a + (1-\alpha)e)v^T, \end{aligned} \tag{44}$$

where $U(\alpha) = \begin{pmatrix} u_1(\alpha) & & & \\ & u_2(\alpha) & & \\ & & \ddots & \\ & & & u_N(\alpha) \end{pmatrix}_{N\times n}$, $V = \begin{pmatrix} e & & & \\ & e & & \\ & & \ddots & \\ & & & e \end{pmatrix}_{n\times N}$, and $v^T = (\frac{n_1}{n}, \frac{n_2}{n}, \cdots, \frac{n_N}{n})$ is a probability vector.

According to this decomposition, in Step 4, we actually only need to compute $A =: U(\alpha)PV$. The corresponding count of multiplications is $O(r)$. Note that we do not need any iteration here, so the complexity of Step 4 is actually much lower than PageRank that will take tens or even hundreds of iterations to converge.

In Step 5, for any starting vector $\xi^{*(0)}(\alpha)$,

$$\begin{aligned} \xi^{*(k)}(\alpha) &= \xi^{*(k-1)}(\alpha)C^*(\alpha) \\ &= \xi^{*(k-1)}(\alpha)U(\alpha)P(\alpha)V \\ &= \xi^{*(k-1)}(\alpha)U(\alpha)(\alpha(P + a\frac{1}{n}e^T) + (1-\alpha)\frac{1}{n}ee^T)V \\ &= \alpha\xi^{*(k-1)}A + (\alpha\xi^{*(k-1)}(\alpha)U(\alpha)a + (1-\alpha))v^T. \end{aligned} \tag{45}$$

Then it is clear that the computational complex of each iteration in Step 5 depends on the number of non-zeroes in A. Because a_{ij} equals to the linear combination of the elements in block P_{ij}, we have $a_{ij} = 0$ when every element is 0 in P_{ij}. Suppose that the average number of sites that a particular website links to is μ, then A has μ non-zeroes in each row and the

number of non-zeroes in A is μN. Considering that for the Web, μ is almost a constant which is tens or so [Bharat et al. 01], we can come to the conclusion that the computational complex of one iteration in Step 5 is $O(N) \ll O(r)$.

5.2 Error Analysis

As one can see, the proposed AggregateRank algorithm is an approximation to PageRankSum. In this subsection, we will discuss the error bound of this approximation.

According to the theory of stochastic complement, the approximation as shown in (17) requires the matrix to be nearly completely decomposable. This condition is well satisfied in real Web applications, because about 75% of the hyperlinks connect pages in the same website [Henzinger et al. 03], and it is reasonable to treat the transition probability matrix $P(\alpha)$ as a nearly completely decomposable matrix.

According to the discussions in Section 5.1.2, $P(\alpha)$ has exactly $N-1$ non-unit eigenvalues that are very close to the unit eigenvalue. Thus, the approximation in (17) has an upper error bound according to [Cho and Meyer], which is determined by the number of pages n, the number of sites N, the size of each site n_i, and the condition number of the coupling matrix $\kappa(C(\alpha))$, the deviation from complete reducibility δ and the eigen structure of the probability transition matrix $P(\alpha)$. We state it explicitly as following theorem.

Theorem 5.1. *If $P(\alpha)$ has exactly $N-1$ non-unit eigenvalues close to the unit eigenvalue, then there are the following error bounds:*

$$\| \frac{\pi_i(\alpha)}{\| \pi_i(\alpha) \|_1} - u_i(\alpha) \|_1 \leq min\{\delta(n_i - 1), 1\}, \tag{46}$$

$$\| C(\alpha) - C^*(\alpha) \|_\infty \leq min\{\delta^2(max_i n_i - 1), 1\}, \tag{47}$$

$$\| \xi(\alpha) - \xi^*(\alpha) \|_1 \leq min\{\delta^2 \kappa(C(\alpha))(max_i n_i - 1), 1\}. \tag{48}$$

From Theorem 4, we can see that the upper error bound of the AggregateRank algorithm principally depends on the scale of the largest website. Evidently, the number of pages in the largest website is much smaller than the size of the Web, i.e. $m = O(n)$. In this regard, we can say that the corresponding error bound is well controlled.

5.3 Experiments

In our experiments, the data corpus is the benchmark data for the Web track of TREC 2003 and 2004, which was crawled from the .gov domain in the year of 2002. It contains 1,247,753 webpages in total.

Before testing our proposed method, we need to partition the Web graph into websites. For this purpose, we follow the rules as below. Because the URLs in the .gov domain are very regular, we can easily decide the website that a webpage belongs to. After removing the $http://$ or $https://$ from the URL, the rest part before the first slash can be considered as the website name. However, because there maybe some subsites, we only use the adjacent word before .gov as the identifier for the site. For example, $http://aaa.bbb.gov/xxxx$ belongs to the website *bbb*.

After this preprocess, we get 731 sites in the .gov dataset. The largest website contains 137,103 web pages while the smallest one contains only 1 page. The distribution of the sizes of all the websites is shown in Figure 3. It nearly follows a power law and is consistent with previous research on the sizes of websites [Albert and Barabasi 02].

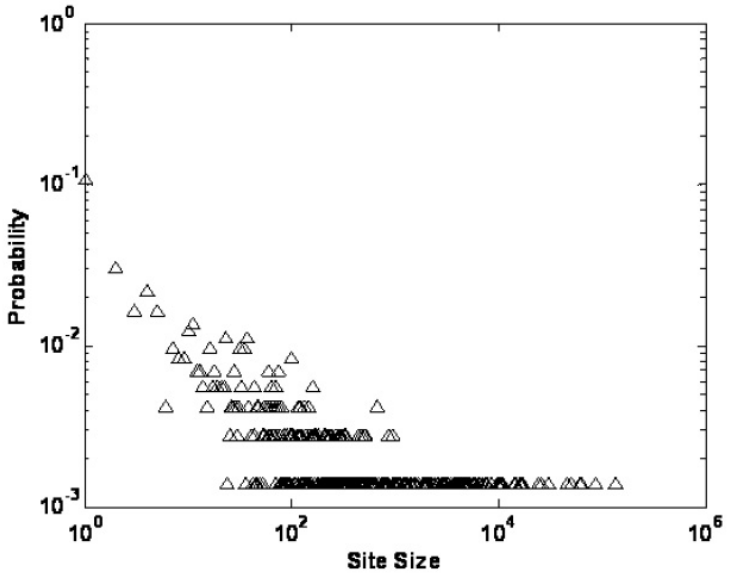

Figure 3: The distribution of the sizes of websites.

In our experiment, we validated whether the proposed AggregateRank algorithm can well approximate PageRankSum. For comparison, we also investigated other two HostRank algorithms, which work on the weighted HostGraph and *naïve* HostGraph respectively [Bharat et al. 01, Dill et al. 01]. The differences between PageRankSum and the algorithms under investigation are shown in Table 1, in terms of Euclidean distance between the rank vectors.

From Table 1, we can see that the AggregateRank algorithm has the best performance: its Euclidean distance from PageRankSum is only 0.0057, while the ranking results produced by the other two algorithms are farther from PageRankSum, with Euclidean distances of 0.1125 and 0.1601 respectively.

Table 1: Performance Evaluation based on Euclidean Distance

Ranking algorithms	Euclidean distance	Max distance in single dimension	Min distance in single dimension
PageRankSum	0.0	0.0	0.0
AggregateRank	0.0057	0.0029	0.000000
Weighted HostRank	0.1125	0.0805	0.000020
naïve HostRank	0.1601	0.1098	0.000007

In addition to the Euclid distance, we also use another similarity measure based on the Kendall's τ distance [Kendall and Gibbons 90] to evaluate the performance of the ranking results. This measure ignores the absolute values of the ranking scores and only counts the partial-order preferences. Therefore, it can better reflect the true ranking performance in real applications. This similarity between two ranking lists s and t is defined as follows.

$$Sim(s,t) = 1 - \frac{K(s,t)}{C_n^2}, \tag{49}$$

where $K(s,t)$ is the Kendall's τ distance, which counts the number of pair-wise disagreements between s and t, and is defined as below.

$$K(s,t) = \mid (i,j) | i < j, s(i) < s(j), t(i) < t(j) \mid . \tag{50}$$

According to the definition, the larger $Sim(s,t)$ is, the more similar two lists are. If the two ranking lists are consistent with each other, their Sim measure is equal to 1.

We list the performance evaluation of the aforementioned algorithm based on Kendall's τ distance in Table 2. From this table, once again we can see that the AggregateRank algorithm is the best approximation to PageRankSum. Furthermore, the advantage of the AggregateRank algorithm over the reference algorithms becomes even more obvious if we look at the top-k ranking results. Actually, we got the top k websites in terms of their PageRankSum, and obtained their order Op. Then, we got their relative orders according to other ranking algorithms, i.e. $O_a, O_w and O_n$ which correspond to the AggregateRank algorithm, the weighted HostRank algorithm and the *naïve* HostRank algorithm respectively. We plot the similarity based on the Kendall's τ distance between these top-k ranking results in Figure 4.

After the comparison on similarity, we compare these ranking algorithms on complexity as well. As discussed in Section 3, the AggregateRank algorithm can converge faster than PageRank. To verify this, we use the L_1-norm of the difference between the current ranking list

Table 2: Performance Evaluation of Ranking Algorithms based on Kendall's τ distance

Ranking algorithms	Sim
PageRankSum	1
AggregateRank	0.9826
Weighted HostRank	0.8428
naïve HostRank	0.8889

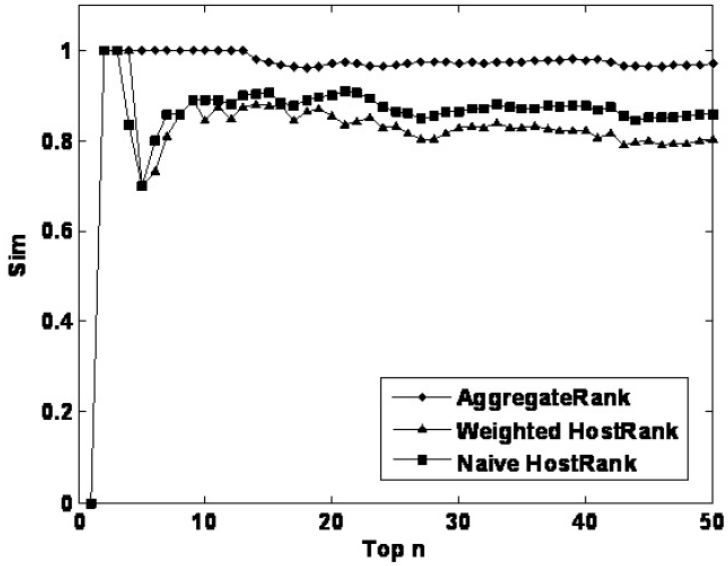

Figure 4: Similarity between PageRankSum and other three ranking results.

and the last one to measure whether the power method converges. When this difference is less than 10^{-3}, we regard the computation as converged and the computation process is terminated. The running time of each algorithm is shown in Table 3.

From Table 3, we can see that the proposed AggregateRank method is faster than PageRankSum, while a little more complex than the HostRank methods. This is consistent with the theoretical analysis in the previous section. The fast speed of AggregateRank mainly comes from the fast convergence speed. And the fast speed of HostRank comes from the low dimension of the HostGraph.

In summary, by taking the effectiveness and efficiency into consideration at the same time, we consider the proposed AggregateRank algorithm as a better solution to website ranking.

Table 3: Comparison of Running Time

Ranking algorithms	Running Time(s)
PageRankSum	116.23
AggregateRank	29.83
Weighted HostRank	0.22
naïve HostRank	0.10

Acknowledgement

The work is partly supported by NSFC, 973 Project and by Science and Practice Fund for Graduate of CAS.

References

[Albert and Barabasi 02] R. Albert and A.-L. Barabasi, *Statistical mechanics of complex networks.* Rev. Mod. Phys. Vol. 74, January 2002.

[Bao and Liu 06] Y.Bao and Y.Liu, *Limit of PageRank with Damping Factor,* Dynamics of Continuous, Discrete and Impulsive Systems, Series B, 13(3), 497-504, 2006.

[Bavulcu et al. 04] H. Bavulcu, S. Vadrevu and S. Nagarajan, *OntoMiner: Bootstrapping Ontologies From Overlapping Domain Specific Web site* . Proceedings of the Thirteenth International World Wide Web Conference, New York, USA, May 2004.

[Bharat et al. 01] K. Bharat, B.-W. Chang, M. Henzinger and M. Ruhl, *Who links to whom: Mining linkage between web sites.* Proceedings of the IEEE International Conference on Data Mining (ICDM '01), San Jose, USA, November 2001.

[Brin et al. 99] S. Brin, L. Page, R. Motwami and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report 1999-0120, Computer Science Cepartmen, Stanford University, Stanford, CA, 1999.

[Cho and Meyer] G. E. Cho and C. D. Meyer, *Aggregation/Disaggregation Methods of Nearly Uncoupled Markov Chains.* $http://meyer.math.ncsu.edu/Meyer/PS_Files/Numcad.ps$

[Despeyroux 04] T. Despeyroux, *Practical Semantic Analysis of Web Sites and Documents.* In Proceedings of the Thirteenth International World Wide Web Conference, New York, USA, May 2004.

[Dill et al. 01] S. Dill, R. Kumar, K. McCurley, S. Rajagopalan, D. Sivakumar and A. Tomkins, *Self-similarity in the Web.* In Proceedings of International Conference on Very Large Data Bases, pp.69-78, Rome, 2001.

[Eiron et al. 04] N. Eiron, K. S. McCurley and J. A. Tomlin, *Ranking the web frontier.* Proceedings of the 13th International World Wide Web Conference (WWW), pp.309-318, New York, NY, USA, 2004. ACM Press.

[Feng et al. 06] Guang Feng, Tie-Yan Liu, Ying Wang, Ying Bao, Zhiming Ma, Xu-Dong Zhang, Wei-Ying Ma, AggrerateRank: Bringing Order to Web Sites, Proceedings of the 29th ACM Conference on Research and Development on Information Retrieval (SIGIR), pp.75-82, Seattle, 2006.

[Gantmacher 59] F. R. Gantmacher, *Matrix Theory*, (Chelsea,1959) Vol.2, Chapter 8.

[Girvan and Newman 02] M. Girvan and M. E. J. Newman, *Community structure in social and biological networks.* Proc. Natl. Acad. Sci. USA, pp.7821-7826, 2002.

[Henzinger et al. 03] M. R. Henzinger, R. Motwani and C. Silverstein, *Challenges in Web Search Engines.* In Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp.1573-1579, 2003.

[Kallenberg 97] O. Kallenberg, *Foundations of Modern Probability,* Springer-Verlag, New York Berlin Herdelberg Tokyo, 1997.

[Kamvar et al. 03] S. Kamvar, T. Haveliwala, C. Manning and G. Golub, *Exploiting the block structure of the web for computing pagerank.* Technical report, Stanford Univ., 2003.

[Kemeny and Snell 76] J. Kemeny and J. L. Snell, *Finite Markov Chains,* Springer-Verlag, New York Berlin Herdelberg Tokyo, 1976.

[Kendall and Gibbons 90] M. Kendall and J. Gibbons, *Rank Correlation Methods.* Edward Arnold, London, 5 edition, 1990.

[Kleinberg 99] J. Kleinberg, *Authoritative sources in a hyperlinked environment,* J.ACM, 46, pp.604-632, 1999.

[Langville and Meyer 04]A. N. Langville and C. D. Meyer, *Deeper inside PageRank.* Internet Mathematics, 1(3), pp.355-400, 2004.

[Lei et al. 04] Y. Lei, Enrico Motta, Domingue, *Modelling Data-Intensive Web Sites with OntoWeaver.* In Proceedings of International Workshop on Web Information Systems Modeling, Riga, Latvia, June 2004.

[Lerman et al. 04] K. Lerman, L. Getoor, S. Minton and C. Knoblock, *Using the Structure of Web Sites for Automatic Segmentation of Tables.* Proceedings of the ACM International Conference on Management of Data, Paris, France, June 2004.

[Meyer 89] C. D. Meyer, *Stochastic complementation, uncoupling markov chains, and the theory of nearly reducible systems,* SIAM Review, 31(2), pp.240-272, 1989.

[Qian and Gong 97] M. Qian, G. Gong, *Theory of Stochastic Processes (the second edition)* (in Chinese), Peking University Press, Beijing, 1997.

[Qin et al. 05] T. Qin, T. Y. Liu, X. D. Zhang, G. Feng and W. Y. Ma, *Subsite Retrieval: A Novel Concept for Topic Distillation.* In Proceedings of 2nd Asia Information Retrieval Symposium, Jeju Island, Kotea, October 2005.

[Stewart 94] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains,* Princeton University Press, Princeton, 1994.

[Wu and Aberer 04] J. Wu and K. Aberer, *Using SiteRank for P2P Web Retrieval.* EPFL Technical Report ID: IC/2004/31, 2004.

Ying Bao, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100080, PR China; Graduate University of the Chinese Academy of Sciences, Beijing,

100049, PR China (ybao@amss.ac.cn)

Gang Feng, Department of Electronic Engineering, Tsinghua University, Beijing, 100084, PR China (fengg03@mails.tsinghua.edu.cn)

Tie-Yan Liu, Microsoft Research Asia, 4F, Sigma Center, No.49, Zhichun Road, Haidian District, Beijing, 100080, PR China (tie-yan.liu@microsoft.com)

Zhi-Ming Ma, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100080, PR China (mazm@amt.ac.cn)

Ying Wang, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, 100080, PR China; Graduate University of the Chinese Academy of Sciences, Beijing, 100049, PR China (wangying@amss.ac.cn)